ACVM - 3. Assignment - Correlation filter tracking

Nejc Ločičnik, 63180183, nl4952@student.uni-lj.si

I. Introduction

In this assignment, we delve into tracking with correlation filters. The implementation follows what we discussed during lectures and what was presented in the article proposing them [1] (mainly the preprocessing used: log and normalization of a patch). We test the available parameters in the simple implementation, an additional template enlargement parameter and the the actual MOSSE filter implementation. The tracker performance is evaluated by running experiments on the VOT2014 sequences [2], using the lite version of the tracking evaluation toolkit.

II. Experiments

A. Basic parameters alpha and sigma

The parameters were found using a manual "smart" grid search. I first did a rough search just to get a feel for what ranges of parameters work best, before focusing on those, to find the optimal result. All together I did about 25 tests for the simple implementation.
α dictates how much the filter is updated. Values from 0.1 to 0.2 seemed to work best. σ is a parameters that regulates the variance, which means a higher sigma allows a higher tolerance to changes. Values from 2 to 3 seemed to work best.
I didn't notice a very clear pattern of how each parameter influences tracker performance. Regarding failures there were situations where $\alpha = 0.9, \sigma = 3$ performed just as good as $\alpha = 0.3, \sigma = 3$. As for the average overlap, I don't think it's a reliable measure since the tracker will re-initialize on the ground truth upon failure, meaning higher failure rates will result in a higher average overlap.
The optimal parameters for the simple implementation were $\alpha = 0.125$ (same as in the article) and $\sigma = 2$. The performance measures of these parameters are shown in table I.

Average Overlap	0.49
Total Failures	59
Average Speed	551

Table I
Evaluation results - basic correlation tracker

B. Enlarging patch size

Here the idea is to test whether increasing the patch size would increase the performance of the tracker. Enlarging the size does potentially improve results, but we need to adjust the α and σ appropriately. On average increasing patch size seemed to increase average overlap, but it also seemed to impact tracker speed, more on this in the section about speed II-D.
The best parameters in this case were $\alpha = 0.15$, $\sigma = 3$ and enlarging factor of 1.3. The performance measures of these parameters are shown in table II.

Average Overlap	0.49
Total Failures	56
Average Speed	340

Table II
Evaluation results - basic with enlarged patches

C. MOSSE filter

The MOSSE filter changes how the correlation filter is updated. Instead of updating and storing the filter, it stores and updates the numerator and denominator separately and then uses them to compute the filter. The updated filter then isn't influenced by the appearance of the old filter. This results in a more stable and accurate representation of the target over time, resulting in an increased performance.

The other change is using the Peak-to-Sidelobe Ratio (PSR) to validate tracking success. For example not doing an online update of the filter, if the PSR detects an occlusion. I implemented it, but I had issues properly thresholding it, so I removed it. I think the results were quite good either way.

Without enlarging patch size, with parameters $\alpha = 0.125$, $\sigma = 2$, the tracker based on the MOSSE filter performed as shown in table III.

Average Overlap	0.51
Total Failures	51
Average Speed	524

Table III
Evaluation results - MOSSE without enlarged patches

While also adding enlarged patch size, with parameters $\alpha = 0.1$, $\sigma = 2$ and enlarging factor of 1.5, the tracker based on the MOSSE filter performed as shown in table IV. As the best performing tracker of those tested, it's results are shown in more detail in table V, where the evaluation measures are shown separately for each sequence in VOT2014.

Average Overlap	0.52
Total Failures	48
Average Speed	300

Table IV
Evaluation results - MOSSE with enlarged patches

In table V we can see that the tracker performed quite poorly on certain sequences. Those sequences are usually a combination of occlusions/size changing (fish1, fish2) or fast movement where the filter doesn't actually fit the tracked object well enough (hand2). This happens because the filter update is weighted more towards the old filter (α is quite low). A lot of these issue get solved if we include an optical flow or a dynamic model that models the movement (which I assume is included in the next assignment).

Figure 1 shows a comparison of a basic tracker with enlarged patches with a tracker using the MOSSE filter. MOSSE filter has better accuracy and robustness.

D. Speed of tracking

Speed of tracking is reported in terms of FPS, which isn't a reliable metric, as it highly varies from computer to computer and highly depends on what else is being done on the computer (watching a youtube video while testing almost halved the FPS for me). The FPS per sequence is reported along other results in table V or averaged out for the whole test set in tables I, II and III, where we can notice that enlarging template size

Sequence	Overlap	Failures	FPS
ball	0.33	0	302
basketball	0.65	3	182
bicycle	0.39	0	886
bolt	0.54	3	365
car	0.44	0	560
david	0.69	0	100
diving	0.37	3	96
drunk	0.32	0	77
fernando	0.33	1	60
fish1	0.35	7	530
fish2	0.36	6	257
gymnastics	0.65	2	156
hand1	0.50	3	248
hand2	0.40	13	251
jogging	0.81	1	343
motocross	0.47	1	37
polarbear	0.46	0	211
skating	0.43	0	298
sphere	0.70	0	114
sunshade	0.77	0	504
surfing	0.80	0	745
torus	0.65	4	442
trellis	0.51	0	148
tunnel	0.32	0	263
woman	0.75	1	323

Table V
Evaluation results of best tracker per sequence

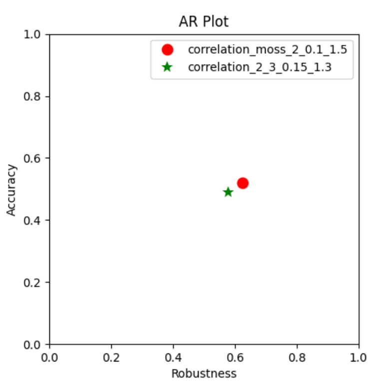

Figure 1. AR plot comparing a basic (green star) and a MOSSE filter tracker (red dot)

reduces the speed of the tracker by quite a bit (550 to 300 FPS), which makes sense as the same operations on bigger matrices require more computation.

In table V we can also see some clear outliers regarding FPS, I am not quite sure why that is. I first thought it might be because of different image sizes or sequence length, but I think it's more likely just FPS being an unreliable metric. I might've been doing something on the laptop while those specific sequences were being tested, which ate up the processor's resources.

Regarding the comparison between the computation time needed to process the initial frame and other frames. I tried measuring the time for an initialization call and a tracking call, but the results were pretty mixed, both usually take 2-5ms, but the tracking call is sometimes the same as initialization and sometimes practically double, so I think it'd be better to evaluate the computation speed from the perspective of operations.

The tracking (processing other frames) needs to load an additional patch and compute localization before constructing a new filter, which follows the same steps as in the initialization. From that point of view we can see that tracking should take more computation than initialization, but the difference shouldn't be significant.

III. Conclusion

In conclusion, the experiments conducted in this assignment explored the effectiveness of correlation filter tracking methods, focusing on parameters such as α and σ, as well as the impact of patch enlargement and the implementation of the MOSSE filter. The results indicated that parameter tuning and template enlargement can improve tracking performance, with the MOSSE filter demonstrating enhanced stability and accuracy. However, further investigation is needed to better understand the relationship between parameter settings and tracking outcomes, as well as the reliability of performance metrics such as FPS in evaluating tracking speed.

References

[1] David S Bolme, J Ross Beveridge, Bruce A Draper, and Yui Man Lui. Visual object tracking using adaptive correlation filters. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 2544–2550. IEEE, 2010.

[2] M. Kristan, J. Matas, A. Leonardis, L. Cehovin, G. Fernandez, T. Vojir, G. Nebehay, and R. Pflugfelder. The visual object tracking vot2014 challenge results. In *Proc. European Conf. Computer Vision*, pages 191–217, 2014.